UNIBUS DISK FORMATTER (FUD31)

USER'S GUIDE

EMULEX
3545 Harbor Boulevard
Costa Mesa, California 92626
(714) 662-5600 TWX 910-595-2521

VX9950904 Rev E
April, 1986

The information in this manual is for information purposes and is subject to change without notice.

Emulex Corporation assumes no responsibility for any errors that may appear in the manual.

Printed in U.S.A.

TABLE OF CONTENTS

BLANK

Section 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

This manual is designed to serve as a guide for those using the Emulex UNIBUS Disk Formatter, FUD31, on Digital Equipment Corporation (DEC) VAX-11 computers. FUD31 is designed to run under the Emulex VAX Monitor, EVM.

This utility is designed for use by qualified installers of Emulex equipment, and thus it assumes that the user has some knowledge of hardware configurations, VAX architecture and terminology, and interpretation of error messages and device register contents.

This document contains two main sections:

Section 1	General Description: This section contains an overview of FUD31, including its functions, distribution media, hardware and software compatibility, and related documentation.
Section 2	Operation: Describes operation of FUD31, including load and start procedures, diagnostic tests, and sample dialog.

1.2 PRODUCT OVERVIEW

The purpose of the FUD31 disk maintenance utility is to prepare new disks for use and/or to maintain integrity of disks already in use.

1.3 DISTRIBUTION MEDIA

The following table lists and describes distribution media for FUD31 and other Emulex VAX diagnostic software.

Emulex P/N	Description
VX9960406	TU58 cassette for VAX-11/750
VX9960506	Eight-inch floppy diskette for VAX-11/780
VX9960910	9-track mag tape for VAX-8600

1.4 COMPATIBILITY

1.4.1 HARDWARE

FUD31 is compatible with DEC VAX-11/730, 11/750, and 11/780 computers. It can be used to test Emulex SC21/V1 and SC31/BX disk controllers.

1.4.2 SOFTWARE

FUD31 is designed to run with the Emulex VAX diagnostic monitor, EVM. For information on EVM, see the EVM user's guide, referenced in subsection 1.5.

1.5 RELATED DOCUMENTATION

The manuals listed in this subsection can be ordered from the following address:

Emulex Corporation
3545 Harbor Blvd.
Costa Mesa, CA 92626
(714) 662-5600 TWX 910-595-2521

Title:	Emulex VAX Monitor (EVM) User's Guide
Publication Number:	VX9950901
Title:	VAX Configuration Utility (IVV000) User's Guide
Publication Number:	VX9950905
Title:	SC21/B1 SC21/V1 SC21/BE SC21/BF SC21/BM (RM02/RM03/RM05 Compatible) Disk Controller Technical Manual
Publication Number:	SC2151001
Title:	SC31/BX (RM02/RM03/RM05/RM80, RP06 Compatible) Disk Controller Technical Manual
Publication Number:	SC3151001

2.1 OVERVIEW

This section describes FUD31 load and start procedures, presents sample configure statements, defines the six tests available with FUD31, and presents sample output.

User input appears in **bold** type, in order to distinguish it from program output. The symbol **<return>** represents the carriage return key.

2.2 LOAD AND START PROCEDURES

The procedure used to invoke EVM varies from one VAX system to another. For a description of EVM bootstrapping procedures, see the Emulex VAX Monitor (EVM) User's Guide (reference given in subsection 1.5).

2.2.1 LOAD PROCEDURE

After the EVM> prompt has appeared on the screen, type the following. (The default filename extension is .EXE.)

```
      EVM>LOAD FUD31<return>
```

The LOAD statement is followed by a SET CONFIGURATION statement, the content of which depends upon the VAX system being used. Sample configure statements are presented in the following subsections.

2.2.2 SAMPLE CONFIGURE STATEMENT FOR VAX-11/730

The following statement refers to a VAX-11/730 with an SC31/BX controller at the standard address and vector (776700, 254) and one Fujitsu M2351 disk drive at drive 0.

```
      EVM>SET CONFIG/CSR:776700/VECTOR:254/DRIVE:0<return>
```

2.2.3 SAMPLE CONFIGURE STATEMENT FOR VAX-11/750

The following statement refers to a VAX-11/750 with an SC31/BX controller at the standard address and vector (776700, 254), one Fujitsu M2351 at drive 0, and UNIBUS adapter UBA0: (base address FC0000):

```
EVM>SET CONFIG/ADAPTER:0/CSR:776700/VECTOR:254/DRIVE:0<return>
```

In the preceding statement, acceptable values for ADAPTER are 0 or 1:

ADAPTER 0	UBA0, FC0000
ADAPTER 1	UBA1, F30000

2.2.4 SAMPLE CONFIGURE STATEMENT FOR VAX-11/780

The following statement refers to a VAX-11/780 with one SC31/BX at the standard address and vector (776700, 254), one Fujitsu M2351 at drive 0, UNIBUS adapter UBA0: (transfer request number 3), and BR (bus request) level 5. SET CONFIGURATION has been abbreviated SE C; for EVM command abbreviations, see the EVM user's guide.

```
EVM>LOAD FUD31.EXE<return>
EVM>SE C/TR:3/ADAPTER:0/CSR:776700/VEC:254/DRIVE:0/BR:5<return>
```

In the preceding statement, acceptable values for TR are 1 through 15, but the range 3 through 6 is the convention:

TR 3	UBA0, 20100000
TR 4	UBA1, 20140000
TR 5	UBA2, 20180000
TR 6	UBA3, 201C0000

BR is optional and needs to be specified only if the BR level of the UNIBUS adapter is other than 4, the standard default. Valid values for BR are 4 through 7.

2.2.5 START PROCEDURE

After typing the LOAD statement and the appropriate SET CONFIGURATION statement, type:

```
EVM>START/TEST:n/PASSES:x<return>
```

where n represents a test number in the range 1 through 6, and **x** represents the desired number of passes in the range 0 through 100.

For FUD31, a test number must be specified; see subsection 2.3 for descriptions of the six available tests. If you do not specify a test number, EVM attempts to run all six tests, starting with test 1. This default will work for many EVM compatible diagnostic programs; FUD31, however, requires that all tests except test 1 be called specifically in order to initialize properly. Therefore, if /TEST is omitted, test 1 runs but test 2 fails with the following error message:

```
ILLEGAL TEST SEQUENCE
```

Although FUD31 tests cannot be chained by default, you can run tests 2 and 3 (format and verify) as a unit by selecting test 1, which combines these two options. See subsection 2.3.1 for details.

The number of passes needs to be specified only if more than one pass is desired; the default value is 1. Valid values for PASSES are 0 through 100, where 0 signifies an infinite number of passes.

2.3 TEST SECTIONS

The six available diagnostic tests are described in the following subsections. After FUD31.EXE has been loaded into memory, information regarding a given test section can also be obtained by typing the following:

```
EVM>HELP/DIAG<return>
```

2.3.1 TEST 1: FORVER (FORMAT AND VERIFY)

The pack format and verify section optionally initializes the bad/skip sector file and performs the format operation followed by the verify test. For detailed descriptions of these two tests, see subsections 2.3.2 and 2.3.3 respectively. This section is provided in order to allow the format and verify sections to be executed as a unit, because test sections cannot otherwise be chained.

During test setup, before any test sections have executed, the operator is prompted to select a 16-bit mask, as explained in subsection 2.3.3.

2.3.2 TEST 2: FORMAT

The pack format utility formats the disk; it is not a test of the data area. It writes sector headers and data fields, using the firmware format feature. Data fields are written with all zeros. The bad/skip sector files are initialized if requested (see subsection 2.3.1).

The operator is given the option to initialize any existing bad sector data, if either Test 1 or Test 2 has been selected and if the pack already contains some bad sector information.

- A Y (yes) response to this option causes all bad (and skipped) sector entries to be deleted.

- A **N** (no) response to this prompt causes the program to preserve all pre-existing bad sector information. The data is saved in memory so that it can be rewritten on the pack after the formatting process is complete.

CAUTION

If the FUD31 disk maintenance utility program is aborted before the format operation has been completed, **all** pre-existing bad sector data is lost.

2.3.3 TEST 3: VERIFY

The surface verification section writes selected data patterns over the entire disk surface, and then performs a write-check operation to locate any bad spots. It adds any errors encountered to the bad/skip sector file, and reformats tracks containing bad spots (if the drive type is RM) so that the defects are reflected by the sector headers.

NOTE

The disk pack must be formatted before the verify section can be run.

Table 2-1 lists the 16 verification data patterns, each containing 16 words, which are used to verify the media. When the program prompts you to select patterns to be used during verify, input a 16-bit mask in which each bit corresponds to a single pattern in Table 2-1: bit 0 = pattern 0, bit 1 = pattern 1, and so on.

Patterns 0 through 3 are selected by default (mask 17 octal), because this combination of patterns usually finds most bad spots on the disk. The time required for the verify test increases with the number of patterns selected.

Table 2-1. Verification Data Patterns (Octal)

Pat-0	Pat-1	Pat-2	Pat-3	Pat-4	Pat-5	Pat-6	Pat-7
000000	165555	133333	177777	000001	177776	000000	000000
000000	133333	165555	177777	000003	177774	000000	010421
000000	165555	133333	177777	000007	177770	000000	021042
000000	133333	165555	177777	000017	177760	177777	031463
000000	165555	133333	177777	000037	177740	177777	042104
000000	133333	165555	177777	000077	177700	177777	052525
000000	165555	133333	177777	000177	177600	000000	063146
000000	133333	165555	177777	000377	177400	000000	073567
000000	165555	133333	177777	000777	177000	177777	104210
000000	133333	165555	177777	001777	176000	177777	114631
000000	165555	133333	177777	003777	174000	000000	125252
000000	133333	165555	177777	007777	170000	177777	135673
000000	165555	133333	177777	017777	160000	000000	146314
000000	133333	165555	177777	037777	140000	177777	156735
000000	165555	133333	177777	077777	100000	177777	167356
000000	133333	165555	177777	177777	000000	000000	177777

Pat-8	Pat-9	Pat-10	Pat-11	Pat-12	Pat-13	Pat-14	Pat-15
052525	007417	026455	077577	000001	177776	172666	077777
052525	007417	026455	077577	000002	177775	155555	137777
052525	007417	026455	077577	000004	177773	172666	157777
125252	170360	151322	077577	000010	177767	155555	167777
125252	170360	151322	077577	000020	177757	172666	173777
125252	170360	151322	077577	000040	177737	155555	175777
052525	007417	026455	077577	000100	177677	172666	176777
052525	007417	026455	077577	000200	177577	155555	177377
125252	170360	151322	077577	000400	177377	172666	177577
125252	170360	151322	077577	001000	176777	155555	177677
052525	007417	026455	077577	002000	175777	172666	177737
125252	170360	151322	077577	004000	173777	155555	177757
052525	007417	026455	077577	010000	167777	172666	177767
125252	170360	151322	077577	020000	157777	155555	177773
125252	170360	151322	077577	040000	137777	172666	177775
052525	007417	026455	077577	100000	077777	155555	177776

2.3.4 TEST 4: UPDATE

The manual update section allows the operator to manually update the bad/skip sector file by adding or deleting sectors.

CAUTION

The update test section must be used with caution, or the integrity of the volume may be jeopardized.

2.3.5 TEST 5: READALL

The readall section reads the entire disk and reports the number of read errors encountered, not including known bad spots. This test does not destroy data on the disk, and it can be run with the disk write protected.

2.3.6 TEST 6: DISPLAY

The display section reads and displays all of the recorded bad/skip sector files from the selected disk.

2.4 SAMPLE DIALOG

In the following sample dialog, only test 1 (format and verify) has been requested by the operator. The number of passes is not specified, so the test is performed once.

When the program asks you to input the pack serial number, you may select any decimal number in the range specified.

The program asks you to input the date only if the drive is an RM80.

```
EVM>START/TEST:1<return>

Emulex VAX-UNIBUS disk maintenance utility  REV X.1  DD-MMM-
YYYY-Time

Unit 0
RM80  (842 Cylinders,  20 Tracks,  48 Sectors)

TEST #1
    ** PACK FORMAT AND VERIFY SECTION **

dd-mmm-yyyy Time
Please enter today's date -

Month [DEC - 1,12]>>> 10<return>

Day [DEC - 1,30]>>> 30<return>

Year [DEC - 84, 1999]>>> 1984<return>

Current date is 10/30/84
```

```
Is this correct [Y,N,(Y)]>>> <return>

Reading bad sector files > Done

The FORVER section will write on the pack and may destroy data.

Do you want to continue [Y,N,(N)]>>> Y<return>

Select patterns to be used during verify [OCT -
1,177777,(17)]>>> <return>
```

Note: Patterns are a bit mask: bit 0 = pattern 0, bit 1 = pattern 1, etc. The default is the worst case four patterns (17).

```
Enter pack serial No.  [DEC - 1,2147483467,(1)]>>> 123<return>

Beginning format on unit 0 at 0:0:0
Format completed at 0:4:9
Rewriting bad sector data.
Beginning verification with pattern 3 at Time
Current No. of bad spots is x
Pattern 3 finished at Time
Beginning verification with pattern 2 at Time
Current No. of bad spots is x
Pattern 2 finished at Time
Beginning verification with pattern 1 at Time
Current No. of bad spots is x
Pattern 1 finished at Time
Beginning verification with pattern 0 at Time
Current No. of bad spots is x
Pattern 0 finished at Time

x new bad spots were found.


SUMMARY REPORT:
TOTAL # ERRORS = 0 (0 SYSTEM, 0 DEVICE, 0 HARD, 0 SOFT)
dd-mmm-yyyy Time
```

BLANK